

# Approximation Algorithms for Directed Multi-Multiway Cut and Directed Multicut Problems

Seyed Naser Hashemi

Department of Mathematics and Computer Science,  
Amirkabir University of Technology  
nhashemi@aut.ac.ir

Ramin Yarinezhad

Department of Mathematics and Computer Science,  
Amirkabir University of Technology  
yarinezhad@aut.ac.ir

October 10, 2016

## Abstract

In this paper, we present approximation algorithms for the directed multi-multiway cut and directed multicut problems. The so called region growing paradigm [1] is modified and used for these two cut problems on directed graphs. By this paradigm, we give for each problem an approximation algorithm such that both algorithms have an approximate factor. The work previously done on these problems need to solve  $k$  linear programming, whereas our algorithms require only one linear programming for obtaining a good approximate factor.

## 1 Introduction

In the following, we first review some of the important cut problems which serve as a background for the problems focused in this paper. The undirected multiway cut problem is defined on an undirected graph  $G = (V, E)$  with a given set  $S = \{s_1, \dots, s_k\} \subseteq V$  of vertices called terminals and a weight function  $c_e$ ,  $e \in E$ . Here the goal is to find the minimum weight subset of edges so that by deleting them, all terminals in  $S$  are disconnected. In other words, there is no any path between any two considered terminals. It is proved that this problem, for  $k \geq 3$ , is NP-hard and MAX SNP-hard, for which a  $2 - 2/k$  factor approximation algorithm is given [5]. In [6], using a geometric relaxation, an algorithm with approximate factor of  $1.5 - 1/k$  is introduced and it is improved to  $1.3438 - \epsilon_k$  in [7].

For directed graphs, the version of directed multiway cut problem is defined. Likewise, given a set of terminals  $S = \{s_1, \dots, s_k\} \subseteq V$  and a weight function  $c_e$ ,  $e \in E$ , we look for a minimum weight subset of edges which their deletions disconnect all directed paths between each pair of terminals. Vazirani and Yannakakis [3] showed that a directed multiway cut problem is also NP-hard and MAX SNP-hard. They introduced an algorithm with a  $2 \log k$  approximate factor. The best known approximation algorithm, presented by Noar and Zosin [4], used a novel relaxation multiway flow to have an approximation algorithm within a factor of 2.

The problem of undirected multicut is another well-known problem defined on undirected graph with a nonnegative cost function  $c_e$ ,  $e \in E$ , and a set of ordered pairs of vertices, namely;  $(s_1, t_1), \dots, (s_k, t_k)$ , which are called source-terminal vertices. In this case, the attempt is to achieve a minimum cost subset of edges so that removing them all sources become inaccessible from their corresponding terminals. For  $k \geq 3$ , it is shown that the problem is NP-hard and MAX SNP-hard [5]. Garg, Vazirani, and Yannakakis [1] give, by the region growing technique, an approximation algorithm with the  $O(\log k)$  approximate factor. In [18] for this problem with more constraints, an approximation algorithm have been proposed with approximation factor  $O(r \log^{3/2} k)$  where  $r$  is a part of input instance.

The directed multicut problem is defined as follows: given an directed graph  $G = (V, E)$ ,  $|V| = n$  with a nonnegative function  $c_e > 0, e \in E$ , and a set of ordered pairs of vertices  $(s_1, t_1), \dots, (s_k, t_k)$ , we find a subset  $F \subseteq E$  with minimum cost function so that their removal from the graph makes each pair disconnected. That is, for any  $i, 1 \leq i \leq k$ , there is no any directed path from  $s_i$  to  $t_i$  in the graph  $G(V, E - F)$ .

Furthermore, if the desire is also to disconnect the paths from  $t_i$  to  $s_i$ , for any  $i, 1 \leq i \leq k$ , we have alternative version of multicut problem called directed symmetric multicut problem.

As shown in [3], for  $k \geq 2$ , the directed multicut problem is NP-hard and MAX SNP-hard. In some papers have been showed another version of this problem is NP-hard [13]. In the literature, most of the works on directed multicut have been focused on the directed symmetric multicut problem [9, 10, 11, 12]. Even, Noar, Schieber and Sudan [11] presented an approximation algorithm with a factor  $O((\log) \log \log k)$ . In general, for a nonsymmetric version, using the technique region growing, an algorithm with the approximate factor  $O(\sqrt{n \log k})$  is given [14]. For the general case, Gupta [15] introduced a simpler algorithm and improved the approximate factor to  $O(\sqrt{n})$ . Both problems above, studied by [14, 15], use a linear programming relaxation to approximate the solution. In the work of Saks, Samorodnitsky, Zosin [17], it is shown that the integrality gap for the linear programming relaxation is  $O(k)$ .

A more general problem on undirected graphs is the multi-multiway cut problem in which the weight function  $w : E \rightarrow \mathbb{R}^+$  and  $k$  sets  $S_1, S_2, \dots, S_k$

are given. Here our aim is to obtain a minimum weight subset of edges whose removal from the graph will disconnect all connections between the vertices in each set  $S_i$ , for  $1 \leq i \leq k$ . For  $K = 1$ , this problem is an undirected multiway cut problem and if  $|S_i| = 2, (1 \leq i \leq k)$ , an undirected multicut problem is obtained. Avidor and Langberg [2] showed that the undirected multi-multiway cut problem is NP-hard and MAX SNP-hard, and by using the region growing technique they could present an approximation algorithm within the factor of  $O(\log k)$ . when the input graph is a tree, In [16] have been showed that this problem is solvable in polynomial time if the number of terminal sets is fixed and in [8] have been presented an approximation algorithm with a factor  $O(\sqrt{k})$ .

A directed version of the above problem is also defined, namely a directed multi-multiway cut problem. Similarly, for this problem a weight function  $w : E \rightarrow \mathbb{R}^+$  on edges and  $k$  sets  $S_1, S_2, \dots, S_k$  are assigned. We attempt to find the minimum weight subset of edges for which their deletions results no paths between any two vertices in each set  $S_i$ , for  $1 \leq i \leq k$ . This problem generalizes the problems of directed multiway cut and directed symmetric multicut (when  $k = 1$  and  $|S_i| = 2$ , respectively).

Since every instance of the directed multiway cut is defined as an instance of the directed multi-multiway cut problem when  $k = 1$ , so the hardness proof for the multiway cut problem implies that the directed multi-multiway cut problem is also NP-hard and MAX SNP-hard.

Note that the problem of directed multi-multiway cut cannot be viewed as an generalization of the undirected multi-multiway cut problem only by replacing each undirected edge by two unparalleled directed edges. For example, consider a tree with a root  $r$ , containing three leaves  $a, b, c$ , and assuming the weight of each edge is equal to one. In this case, we get the optimal value,  $OPT = 2$ , whereas substituting each edge by two directed edges gives  $OPT = 3$ , and this proves our claim that two problems above are not equivalent.

As described above, it is clear that the problems of directed multicut and directed multi-multiway cut can be approximated by a factor  $O(k)$ . But for each of these problems, we require  $k$  linear programming to be solved in order to obtain the desired approximation solution. In this paper, we show that we can achieve the same result, i.e. an approximation with the factor  $O(k)$ , by solving only one linear programming. For this, the so called paradigm of region growing, introduced in [1] for undirected cut problems, is modified so that it can be useful for approximate solution of the multicut and multi-multiway cut problems on directed graphs..

The rest of this paper is organized as follows: In section 2 we present a linear programming relaxation for the Directed Multi-Multiway Cut problem which is used in [2] and [3]. Section 3 contains necessary definitions and lemmas for algorithm Directed Multi-Multiway cut which proposed in section 4. Directed Multicut Algorithm presented in section 5 and Conclusion

is brought in section 6.

## 2 Linear Programming Relaxation for the Directed Multi-Multiway Cut

We define a decision variable  $x(e)$  for each edge  $e$  which is as following: if  $e$  belongs to directed multi-multiway cut,  $x(e) = 1$ , otherwise  $x(e) = 0$ . The purpose is to find a directed multi-multiway cut with minimum cost which disconnects every directed path between two vertices in a group. We call that the set of all directed paths between any two vertices belongs to a group, with  $P$ . An integer program for the problem is given by:

$$\begin{aligned} & \text{minimize} && \sum_{e \in E} w(e)x(e) \\ & \text{subject to} && \sum_{e \in p} x(e) \geq 1, \quad \forall p \in P \\ & && x(e) \in \{0, 1\}, \quad \forall e \in E \end{aligned}$$

By relaxing this IP we obtain the following linear programming relaxation:

$$\begin{aligned} & \text{minimize} && \sum_{e \in E} w(e)x(e) \\ & \text{subject to} && \sum_{e \in p} x(e) \geq 1, \quad \forall p \in P \\ & && x(e) \geq 0, \quad \forall e \in E \end{aligned}$$

In this LP, there is a constraint for each path. On the other hand, we may have exponential number of paths with respect to the input size and as a result, exponential number of constraints. Nevertheless we can solve this LP in polynomial time, using ellipsoid algorithm. For this LP, the separation oracle operates as following: we get an answer  $x$  and assume that the length of each edge  $e$  is equal to  $x(e)$ . Then we find the shortest directed path between two edges which are needed to be disconnected from one another, for example  $(u, v)$ , If the shortest path between  $u$  and  $v$  (either  $v \rightarrow u$  or  $u \rightarrow v$ ), is more than 1, then this constraint  $\sum_{e \in p} x(e) \geq 1$  is true for all paths between this two vertices. So this LP can be solved in polynomial time.

## 3 Definitions and Lemmas

To round the solution of this LP and obtain a directed multi-multiway cut, we use the region growing technique [1, 2]. Noticing that definitions in [1, 2] are related to undirected graphs while definitions presented here, are related to directed graphs.

We define a distance on edges, Assume  $x$  is an optimal solution for LP, Let  $x(e)$  be the length of edge  $e$ . The distance between two vertices  $u$  and  $v$  (either  $v \rightarrow u$  or  $u \rightarrow v$ ) which is defined based on  $x(e)$ , is the length of a shortest path between them. We represent this shortest path with  $dist(u, v)$ . If there is no directed path between two vertices  $u$  and  $v$ , the value of  $dist(u, v)$  is equal to the value of the shortest path in graph, regardless of the direction of edges. Assume that,  $1 \leq i \leq k, 1 \leq j \leq |S_i|$ , We define:

$$B_x(s_{ij}, r) = \{v \in V : dist(s_{ij}, v) \leq r\}$$

$B_x(s_{ij}, r)$  is an area like a ball with center  $s_{ij}$  and radius  $r$ .

Assume that  $\delta(s)$  is the set of all edges which only one of their heads is in the set  $s$ . For a given radius  $r$ , we define  $wt(\delta(B_x(s_{ij}, r)))$  is the sum of the weights of all edges which one of their heads is in  $B_x(s_{ij}, r)$  :

$$wt(\delta(B_x(s_{ij}, r))) = \sum_{e \in \delta(B_x(s_{ij}, r))} w(e)$$

The same as [2], we define an upper bound on the weight of directed edges which their one head only is inside these balls:

$$c_i(r) = \sum_{j=1}^{|S_i|} wt(\delta(B_x(s_{ij}, r)))$$

Assume that the product of  $w(e)x(e)$  is equal to the volume of edge  $e$ . Thus the answer of LP is the value of edges with minimum volume such that  $dist(u, v) \geq 1$  for every  $u$  and  $v$  in which  $u$  and  $v$  are in the same group and there is a path between them.(either  $v \rightarrow u$  or  $u \rightarrow v$ ). Let  $x$  is an optimal solution for LP and  $V^* = \sum_{e \in E} w(e)x(e)$  be the volume of all edges, indeed  $V^*$  is the optimal value of LP. We know that  $V^* \leq OPT$  such that  $OPT$  is the optimal value for IP. Let  $v_i(r)$  is defined as follows:

$$v_i(r) = \beta V^* + \sum_{j=1}^{|S_i|} \left( \sum_{\substack{e=(u,v) \in E \\ u,v \in B_x(s_{ij}, r)}} w(e)x(e) + \sum_{\substack{e=(u,v) \in E \\ u \in B_x(s_{ij}, r) \\ v \notin B_x(s_{ij}, r)}} w(e)(r - dist(s_{ij}, u)) \right)$$

Such that  $\beta > 0$  and is independent from  $r$ . We have to notice that an edge may appear in  $c_i(r)$  more than once. That means we may have  $\delta(B_x(s_{ij}, r)) \cap \delta(B_x(s_{ij'}, r)) \neq \emptyset$  (for  $1 \leq j \neq j' \leq |S_i|$ ), thus  $c_i(r)$  is an upper bound on the cut.

Lemma 2 says in directed graphs, we can always find a radius  $r < \frac{1}{2}$ , such that the cost  $v_i(r)$  is an upper bound for  $c_i(r)$ .

**Lemma 1:** The function  $v_i(r)$  is differentiable in  $(0, \infty)$  except some finite number of points. The derivative of this function is  $c_i(r)$ .

**Proof:** The function  $v_i(r)$  is not differentiable in points which the value of function  $B_x(s_{ij}, r)$  changes. The function  $B_x(s_{ij}, r)$ , changes for the values of  $r$  in which there is a vertex  $v$  such that  $dist(s_{ij}, r) = r$ . Thus the number of points in which the function  $v_i(r)$  is not differentiable, is finite. Besides, according to the definition done for the function  $v_i(r)$ , the derivative of this function is  $c_i(r)$ . ■

**Lemma 2:** Let  $x$  is a feasible solution for LP, for every  $s_{ij}$  there is a  $r < \frac{1}{2}$  and at least an  $\alpha(\alpha > 0)$  such that the following inequality is true:

$$c_i(r) \geq \alpha v_i(r)$$

The proof of this lemma is brought in the rest of the paper. We first present the algorithm which is using this lemma.

## 4 Approximation Algorithm for Directed Multi-Multiway Cut

The algorithm solves the LP first and finds the optimal solution  $x$ . Then the algorithm enters to a repetition loop and till there exists a path between two vertices in a group, the algorithm works as following. Assume that the set  $S_i$  is chosen in this iteration. In the beginning it finds a  $r$  which satisfies the inequality of lemma 2, then it finds the set of balls with the center of vertices inside the  $S_i$  and with the radius of  $r$  and puts the edges been cut by these balls in the answer set and then deletes these edges from graph. Now the algorithm checks whether there exists a condition in which two vertices in another group that there is a path between them, exist in one of the balls or not. In existence condition, only the vertices in the center of balls and incident edges with these vertices will be deleted from graph. Else all of the vertices in balls and incident edges with these vertices will be deleted from graph. The algorithm 1 is shown in following.

**Lemma 3:** The algorithm return a Directed Multi-Multiway Cut.

**Proof:** No vertex in a same group with the center of balls is inside the balls. Because the radius of each area is smaller than  $\frac{1}{2}$ . The only condition that may make a problem, is when two vertices which are in another group and there is a path between them are in one ball. In this case only the central vertices and the edges crossover with these vertices will be deleted from the graph. So the path between two vertices which are the member of

---

**Algorithm 1** Approximation Algorithm for Directed Multi-Multiway Cut

---

**Result:** A Directed Multi-Multiway Cut

$F \leftarrow \emptyset$

Solve LP and get optimal solution  $x$

**while** there is a path between  $s_{ij} \in S_i$  and  $s_{ij'} \in S_i$  (where  $1 \leq i \leq k, 1 \leq j \leq |S_i|$ ) **do**

    Find  $r_i$  such that  $c_i(r_i) \leq \alpha v_i(r_i)$

    Add  $\bigcup_{j=1}^{|S_j|} \delta(B_x(s_{ij}, r_i))$  to  $F$

    Remove  $\bigcup_{j=1}^{|S_j|} \delta(B_x(s_{ij}, r_i))$  from the graph

**if**  $\bigcup_{j=1}^{|S_j|} B_x(s_{ij}, r_i)$  contain two vertex  $u, v$  such that  $u, v \in B_x(s_{ij}, r_i)$  (where  $1 \leq j \leq |S_i|$ ) AND  $u, v \in S_m$  (where  $1 \leq m \leq k$ ) AND (there is a path between  $u$  and  $v$ ) **then**

        Remove  $s_{ij}$  (where  $1 \leq j \leq |S_i|$ ) and incident edges with it from the graph

**else**

        Remove  $\bigcup_{j=1}^{|S_j|} B_x(s_{ij}, r_i)$  and incident edges with it from the graph

**end**

$\forall l \in 1, \dots, k \ S_l \leftarrow S_l \cap V$

**end**

Return  $F$

---

another group will not be deleted from the graph and in the next iteration, at least one of edges of this path will be in the answer set of algorithm.

**Theorem 1:** Algorithm 1 is an  $(\alpha(1 + \beta)k)$ -approximation algorithm for Directed Multi-Multiway Cut.

**Proof:** According to lemma 2, we have  $c_i(r) \leq \alpha v_i(r)$  for every  $1 \leq i \leq k$ . Thus  $\sum_{i=1}^k c_i(r) \leq \alpha \sum_{i=1}^k v_i(r)$ . Besides, according to the definition of  $v_i(r)$  and algorithm method, we have  $\sum_{i=1}^k v_i(r) \leq (kV^* + k\beta V^*)$ . Thus:

$$F \leq \sum_{e \in F} w(e) = \sum_{i=1}^k c_i(r) \leq \alpha \sum_{i=1}^k v_i(r) \leq \alpha(1 + \beta)kV^* \leq \alpha(1 + \beta)kOPT$$

■

#### 4.1 The proof of lemma 2 and finding the best values for $\alpha$ and $\beta$

**The proof of lemma 2:** We use the contradiction method. Assume that for every value of  $r < \frac{1}{2}$  and every  $\alpha(\alpha > 0)$  we have  $c_i(r) > \alpha v_i(r)$ . Thus

$$\begin{aligned} c_i(r) &> \alpha v_i(r) \\ \int_0^{\frac{1}{2}} \frac{c_i(r)}{v_i(r)} dr &> \alpha \int_0^{\frac{1}{2}} dr \end{aligned}$$

According to lemma 1, the function  $v_i(r)$  is not differentiable at only finite number of point. We call these points  $r_0 = 0 \leq r_1 \leq \dots \leq r_l \leq r_{l+1} = \frac{1}{2}$ . Thus we can be write:

$$\begin{aligned} \int_0^{\frac{1}{2}} \frac{1}{v_i(r)} \left( \frac{dv_i(r)}{dr} \right) dr &= \sum_{j=0}^l \int_{r_j}^{r_{j+1}} \frac{1}{v_i(r)} \left( \frac{dv_i(r)}{dr} \right) dr \\ &\leq \ln v_i\left(\frac{1}{2}\right) - \ln v_i(0) \end{aligned}$$

Since  $v_i(r)$  is an increasing function, we have

$$\leq \ln v_i\left(\frac{1}{2}\right) - \ln v_i(0)$$

And  $v_i(0) = \beta V^*$  and  $v_i(\frac{1}{2}) \leq \beta V^* + V^*$  thus we have:



$$\begin{aligned}
\ln\left(\frac{\beta V^* + V^*}{\beta V^*}\right) &\geq \ln\left(\frac{v_i(\frac{1}{2})}{v_i(0)}\right) > \frac{\alpha}{2} \\
\ln\left(\frac{\beta + 1}{\beta}\right) &> \frac{\alpha}{2}
\end{aligned} \tag{*}$$

To reach a contradiction we have to choose values for  $\alpha$  and  $\beta$  which the inequality (\*) would not be true. On the other hand, the approximation factor of algorithm is dependent to these two parameters directly. So we have to choose the minimum value for  $\alpha$  and  $\beta$ .

Indeed to find the best value for  $\alpha$  and  $\beta$ , we should solve the following non linear program:

$$\begin{aligned}
&\text{minimize } \alpha(1 + \beta) \\
&\text{subject to } \ln\left(\frac{\beta + 1}{\beta}\right) \leq \frac{\alpha}{2} \\
&\alpha, \beta > 0
\end{aligned}$$

We solved this problem using Matlab software and found the optimal value of  $\alpha$  and  $\beta$ . These values are:  $\alpha = 0.1$  and  $\beta = 20.504$ . If we put these values in the inequality (\*), the contradiction is reached and lemma 2 is proved. Using these values for  $\alpha$  and  $\beta$ , the algorithm is an approximation algorithm with the factor  $(2.1504)k$  for Directed Multi-Multiway cut problem.

■

## 5 Approximation Algorithm for Directed Multi-cut

Similar to LP presented in the previous section can be provided a LP for the Directed Multicut problem. We define a decision variable  $x(e)$  for each edge  $e$  which is as following: If  $e$  belongs to directed multicut,  $x(e) = 1$ , Otherwise  $x(e) = 0$ . The purpose is to find a directed multicut with minimum weight which cuts each directed path from  $s_i$  to  $t_i$  for  $1 \leq i \leq k$ . We represent the set of all directed paths from  $s_i$  to  $t_i$  for  $1 \leq i \leq k$  with  $P$ . An linear programming for the problem is as follows:

$$\begin{aligned}
&\text{minimize } \sum_{e \in E} w(e)x(e) \\
&\text{subject to } \sum_{e \in p} x(e) \geq 1, \quad \forall p \in P \\
&\quad \quad \quad x(e) \geq 0, \quad \forall e \in E
\end{aligned}$$

In this LP, there is a constraint for each path. On the other hand, we may have exponential number of paths with respect to the input size and as a result, exponential number of constraints. Nevertheless we can solve this LP in polynomial time, using ellipsoid algorithm. For this LP, the separation oracle operates as following: we get an answer  $x$  and assume that the length of each edge  $e$  is equal to  $x(e)$ . Then we find the shortest directed path from  $s_i$  to  $t_i$  for  $1 \leq i \leq k$ . If the shortest path from  $s_i$  to  $t_i$ , is more than 1, then constraint  $\sum_{e \in p} x(e) \geq 1$  is true for all paths from  $s_i$  to  $t_i$ . So this LP can be solved in polynomial time.

We provide direct version of definitions those used in region growth technique in [1]. We define a distance on edges, assume that  $x$  is an optimal solution for LP, let  $x(e)$  be the length of the edge  $e$ . We show the shortest path from  $u$  to  $v$  which we defined based on  $x(e)$  with  $dist(u, v)$ . Assume that  $1 \leq i \leq k$ , then we have  $dist(s_i, t_i) \geq 1$ . If there is no a directed path from  $u$  to  $v$ , the value of  $dist(u, v)$  is the shortest distance from  $u$  and  $v$  in the graph, without noticing the direction of edges. Now we define:

$$B_x(s_i, r) = \{v \in V : dist(s_i, v) \leq r\}$$

$B_x(s_i, r)$  is an area like a ball with center  $s_i$  and radius  $r$ .

Assume that the product of  $w(e)x(e)$  is equal to the volume of edge  $e$ . Thus the answer of LP is the value of edges with minimum volume such that  $dist(s_i, t_i) \geq 1$  for  $1 \leq i \leq k$ . Assume that  $x$  is an optimal solution for LP. Let  $V^* = \sum_{e \in E} w(e)x(e)$  be the volume of all edges, indeed let  $V^*$  be the optimal value of LP. We know that  $V^* \leq OPT$  such that  $OPT$  is the optimal value for IP. Assume that  $v_x(s_i, r)$  is the volume of all edges in maximum distance between  $r$  and vertex  $s_i$ , plus an additional value of  $\beta V^*$  ( $\beta > 0$ ). Therefore  $v_x(s_i, r)$  is defined as follows:

$$v_x(s_i, r) = \beta V^* + \sum_{\substack{e=(u,v) \in E \\ u,v \in B_x(s_i, r)}} w(e)x(e) + \sum_{\substack{e=(u,v) \in E \\ u \in B_x(s_i, r) \\ v \notin B_x(s_i, r)}} w(e)(r - dist(s_i, u))$$

Assume that  $\delta(s)$  is the set of all edges which only one of their heads is in the set  $s$ . For a given radius  $r$ , we define:

$$wt(\delta(B_x(s_i, r))) = \sum_{e \in \delta(B_x(s_i, r))} w(e)$$

The following lemma demonstrate that in directed graphs, we can always find a radius  $r < \frac{1}{2}$ , such that the cost  $v_x(s_i, r)$  is an upper bound for  $wt(\delta(B_x(s_i, r)))$ .

**Lemma 3:** The function  $v_x(s_i, r)$  is differentiable in  $(0, \infty)$  except some finite number of points. The derivative of this function is  $wt(\delta(B_x(s_i, r)))$ .

**Lemma 4:** Assume that  $x$  is a feasible solution for LP. For every  $s_i$  there is a  $r < \frac{1}{2}$  and at least an  $\alpha$  ( $\alpha > 0$ ) such that the following inequality is true:

$$wt(\delta(B_x(s_i, r))) \leq \alpha v_x(s_i, r)$$

The proof of lemma 3 and lemma 4 is similar to proof of lemma 1 and lemma 2. For simplicity we assume that  $wt(r) = wt(\delta(B_x(s_i, r)))$  and  $v(r) = v_x(s_i, r)$ .

The algorithm first solves LP and finds the optimal answer  $x$ . We consider the value of answers  $x$  as the length of edges in the graph. In every iteration, the algorithm finds a pair which there is a path between them and also finds an area with a radius that satisfies the conditions in lemma 4. Then we put the cutting edges which are created by the area in the set  $F$ . If this area is included another pair  $(s_j, t_j)$  and there is a path from  $s_j$  to  $t_j$ , then we remove only the central vertex of area. So we can cut the path between  $s_j$  and  $t_j$  on the next iteration. The algorithm 2 is pseudo code of this algorithm.

---

**Algorithm 2** Approximation Algorithm for Directed Multicut

---

**Result:** A Directed Multicut

$F \leftarrow \emptyset$

Solve LP and get optimal solution  $x$

**while** there is a path between  $s_i$  to  $t_i$  (where  $1 \leq i \leq k$ ) **do**

    Grow a region  $S = B_x(s_i, r)$  until  $wt(r) \leq \alpha v(r)$

    Add  $\delta(S)$  to  $F$

    Remove  $\delta(S)$  form the graph

**if** ( $S$  contain a pair  $(s_j, t_j)$  (where  $1 \leq j \leq k$ )) **AND** (there is a path form  $s_j$  to  $t_j$ ) **then**

        Remove  $s_i$  and incident edges with  $s_i$  form the graph

**else**

        Remove  $S$  and  $\delta(S)$  form the graph

**end**

**end**

Return  $F$

---

**Lemma 5:** The algorithm returns a directed multicut.

**Proof:** Consider the ball  $B_x(s_i, r)$ . The center of this ball is  $s_i$ . Vertex  $t_i$  cannot be in this ball Because the radius of the ball is less than  $\frac{1}{2}$  ( $r < \frac{1}{2}$ ).

Besides, according to LP constraint, we know that the distance between every pair  $(s_i, t_i)$  ( $1 \leq i \leq k$ ) should be more than 1. The only case that may make some problems is that a pair  $(s_j, t_j)$  which there is a path from  $s_j$  to  $t_j$ , stands in this area. In this case, the algorithm only removes the central vertex  $s_i$  from graph, thus the pair  $(s_j, t_j)$  and also the path between these vertices are still in the graph. In the rest of iterations, the algorithm would make them disconnected.

■

**Theorem 2:** The Algorithm is an  $O(k)$ -approximation algorithm for the Directed Multicut problem.

**Proof:** We demonstrate the set of vertices in the ball  $B_x(s_i, r)$  with  $B_i$  (We assume that  $B_i = \emptyset$  when no ball is selected for vertex  $s_i$ ). We also demonstrate the set of cut edges for  $B_i$  with  $F_i$ . It means  $F_i$  is equal to  $\delta(B_i)$ . Thus we have  $F = \bigcup_{i=1}^k F_i$ . Assume that  $V_i$  is equal to the volume of all edges which are in the ball  $B_i$  and also the volume of edges which their one head is in  $B_i$ . According to this definition we have  $V_i \geq v_x(s_i, r) - \beta V^*$ , because  $V_i$  includes the volume of all edges in  $F_i$ . But  $v_x(s_i, r)$  is included only some part of these edges and an addition value  $\beta V^*$ . According to Lemma 4 and the choice done for  $r$  in the algorithm, we have  $wt(F_i) \leq \alpha v_x(s_i, r) \leq \alpha(V_i + \beta V^*)$ . We know that the algorithm may not remove the edges which are among the  $B_i$  vertices in this iteration in which  $s_i$  is chosen. Thus an edge may belong to more than one area but we would have at most  $k$  areas. Therefore we have  $\sum_{i=1}^k V_i \leq kV^*$ . So:

$$\sum_{e \in F} w(e) = \sum_{i=1}^k wt(F_i) \leq \alpha \sum_{i=1}^k (V_i + \beta V^*) \leq \alpha(1 + \beta)kV^* \leq \alpha(1 + \beta)kOPT$$

■

Similar to section 4 optimal value for  $\alpha$  is 0.1 and  $\beta$  is 20.504, So This algorithm is an  $(2.1504)k$ -approximation algorithm for Directed Multicut problem.

## 6 Conclusions

In this paper, we design approximation algorithms for the directed multiway cut and directed multicut problems using the region growing technique [1, 2]. By this paradigm, we give for each problem an  $O(k)$ -approximation algorithm. The work previously done on these problems need to solve  $k$  linear programming, whereas our algorithms require only one linear programming. Both algorithms use the same linear programming relaxation. A question of interest is to find the integrality gap of the linear programming relaxation for this problem.

## References

- [1] Garg, Naveen, Vijay V. Vazirani, and Mihalis Yannakakis. "Approximate max-flow min-(multi) cut theorems and their applications." *SIAM Journal on Computing* 25.2 (1996): 235-251.
- [2] Avidor, Adi, and Michael Langberg. "The multi-multiway cut problem." *Theoretical Computer Science* 377.1 (2007): 35-42.
- [3] Garg, Naveen, Vijay V. Vazirani, and Mihalis Yannakakis. "Multiway cuts in directed and node weighted graphs." *Automata, Languages and Programming*. Springer Berlin Heidelberg, 1994. 487-498.
- [4] Naor, Joseph, and Leonid Zosin. "A 2-approximation algorithm for the directed multiway cut problem." *SIAM Journal on Computing* 31.2 (2001): 477-482.
- [5] Dahlhaus, Elias, et al. "The complexity of multiway cuts." *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*. ACM, 1992.
- [6] Clinescu, Gruia, Howard Karloff, and Yuval Rabani. "An improved approximation algorithm for multiway cut." *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998.
- [7] Karger, David R., et al. "Rounding algorithms for a geometric embedding of minimum multiway cut." *Mathematics of Operations Research* 29.3 (2004): 436-461.
- [8] Zhang, Peng, Daming Zhu, and Junfeng Luan. "An approximation algorithm for the Generalized k-Multicut problem." *Discrete Applied Mathematics* 160.7 (2012): 1240-1247.
- [9] Even, Guy, et al. "Divide-and-conquer approximation algorithms via spreading metrics." *Journal of the ACM (JACM)* 47.4 (2000): 585-616.
- [10] Leighton, Tom, and Satish Rao. "An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms." *Foundations of Computer Science, 1988., 29th Annual Symposium on*. IEEE, 1988.
- [11] Even, Guy, et al. "Approximating minimum feedback sets and multicuts in directed graphs." *Algorithmica* 20.2 (1998): 151-174.
- [12] Klein, Philip N., et al. "Approximation algorithms for Steiner and directed multicuts." *Journal of Algorithms* 22.2 (1997): 241-269.

- [13] Bang-Jensen, Jrgen, and Anders Yeo. "The complexity of multicut and mixed multicut problems in (di) graphs." *Theoretical Computer Science* 520 (2014): 87-96.
- [14] Cheriyan, Joseph, Howard Karloff, and Yuval Rabani. "Approximating directed multicuts." *Combinatorica* 25.3 (2005): 251-269.
- [15] Gupta, Anupam. "Improved approximation algorithm for directed multicut." *SODA 2003* (2003): 454-455.
- [16] Kanj, Iyad, et al. "Improved parameterized and exact algorithms for cut problems on trees." *Theoretical Computer Science* 607 (2015): 455-470.
- [17] Saks, Michael, Alex Samorodnitsky, and Leonid Zosin. "A lower bound on the integrality gap for minimum multicut in directed networks." *Combinatorica* 24.3 (2004): 525-530.
- [18] Chuzhoy, Julia, et al. "Approximation algorithms and hardness of the k-route cut problem." *ACM Transactions on Algorithms (TALG)* 12.1 (2016): 2.